Installation and post installation problems

Describes the solutions to common problems with the installation and post installation.

---

1. Possible Problems with Installation

See the installation documentation for general problems with the installation. I have found that ioapi specific problems are usually due to problems with Fortran and the Fortran libraries. For Fortran errors, check the following logs: netcdf, ioapi, and pyIoapi. Here are some possible problems and their solutions:

a. Fortran turned off in netCDF build

This is probably due to the Fortran compiler not being picked up by the netCDF configuration file. This seems to happen when you are using gfortran on a Mac 10.4. The solution is to explicity set the Fortran compiler variable. For example, if you were doing an express build with gfortran:

```
$ FC=gfortran ./express_install ~/ --enable-ioapi
```

b. pyIoapi build fails because multiple Fortran compilers

If you have multiple Fortran compilers, your build may fail because different compilers are being used to build components of the pyIoapi shared library. For example, if you had two Fortran compilers, pgf and g77, and you wanted to build everything with g77, then you should run:

```
$FC=g77 ./express_install ~/ --enable-ioapi
```

This will build all of the dependent libraries as g77. But, the pyIoapi linker may default to pgf90. If this happens, then you need to explicitly tell pyIoapi to build using g77. Change directory to <CDAT_SRC_DIRECTORY>/contrib/pyIoapi/build/pyIoapi. Edit the configuration file setup.cfg by uncommenting the line:

```
#fcompiler=gnu
```

Remove the build directory in this directory. Then rerun pyIoapi install:

```
$ <CDAT_PYTHON>/python setup.py install --prefix=~/
```

If you are using the system python, then run this as root. If you are installing it in the python directory, then you don't use the prefix option.

c. multiple contrib packages fail, g77 from fink

On a Mac with g77 from fink, many of the contrib packages might fail with an error about not finding "−lg2c". This is due to a linking error when using pyfort. My simple solution was to make a link in /usr/lib:

```
# ln -s /sw/lib/libg2c.a /usr/lib/libg2c.a
```

Contents

---

2. Possible Problems Post Installation

a. can't find module

```
Traceback (most recent call last):
  File "<stdin>", line 1, in ?
ImportError: No module named ioapiTools
```

This error means that the module is not in the python path. If your using the express install, make sure that your using the CDAT installed python interpeter. It might require you to give the full path to python. If you are using your own version of python, but installed the modules in a seperate directory, then you need to tell the python interpreter where the modules are located. For example, if you are using the standard python, but decided to install cdat in ~/, then you should set the PYTHONPATH environment variable to the location of the CDAT modules:

```
$ export PYTHONPATH=~/lib/python2.4/site-packages
```

b. subset or coordinate conversion error, can't find library

If you get this error:

```
Traceback (most recent call last):
  File "./unit_ioapiTools.py", line 447, in testCdmsmetaLatlonAxes
    self.failUnless(cdmsM.latlonAxes(o3.ioM), \
  File "/home/azubrow/lib/python2.4/site-packages/ioapiTools/ioapiTools.py", line 3398, in latlon
    coordOut = coordConv(ioM, coordLst, proj2llFlag)
  File "/home/azubrow/lib/python2.4/site-packages/ioapiTools/ioapiTools.py", line 4042, in coordC
    LL2Proj = osr.CoordinateTransformation(projLonLat, proj)
  File "/home/azubrow/lib/python2.4/site-packages/osr.py", line 679, in __init__
    raise ValueError, gdal.GetLastErrorMsg()
ValueError: Unable to load PROJ.4 library (libproj.so), creation of
OGRCoordinateTransformation failed.
```

or something about unable to load gdal library, it means that the dynamic linker can't find the shared library. The solution is to tell the linker where that library is located. For example, if you built cdat in your home directory, you would set the LD_LIBRARY_PATH to include that lib directory. For a bash system:

```
$ export LD_LIBRARY_PATH=~/lib:$LD_LIBRARY_PATH
```

. Contents